Relationship between mature software engineering practices and agility practices

Alvaro Ruiz de Mendarozqueta[1], Fabio O. Bustos[2] and Pedro E. Colla[3]

[1] aruizdemendarozqueta@gmail.com, [2]fabio.oscar.bustos@gmail.com, [3] colla.pedro@uader.edu.ar
Universidad Tecnológica Nacional – Regional Córdoba[1]
Córdoba – Córdoba - Argentina

Abstract. This paper reports on research work on Argentinean software development organizations. The analysis provides insights on the profile of the companies regarding the usage of agile methods and software engineering practices trends, their motivations, and drivers. The conclusions can be used to understand what drivers facilitate the understanding of bonds between both in order to increase their competitiveness in domestic and off-shore markets.

Keywords: Survey software organizations in Argentina. Agile, SCRUM, Software Engineering

1 INTRODUCTION

A research effort aiming to understand how organizations perceive the relationship between agile methodologies and traditional software engineering practices has been carried out by the authors (Colla, Bustos, & Ruiz de Mendarozqueta, Agile and software engineering, an invisible bond, 2020), an experiment using the data gathered by means of a survey among software organizations in Argentina, and the analysis of its results throw some additional light on the subject. In these software organizations, activities are executed for the development of standard products and customized implementations, update and maintenance of existing products, as well as embedded applications for electronic devices. Software size and complexity are increasing rapidly and the total software staff is growing continuously; still, most of the work is performed at Small and Medium Enterprises (SMEs) organizations (OPSSI, 2016). Previously published efforts (Colla, Ruiz de Mendarozqueta, & Bustos, Agile in practice, a systemic approach, 2020) from the authors have been focused on building a preliminary discussion about the importance of addressing best, well-established practices from software engineering-based methodologies when adopting agile methods and premises. Also, a detailed discussion on how the agile and software engineering concepts are strongly bonded, even if this relationship is not often highlighted by the bibliography, has been made (Colla, Bustos, & Ruiz de Mendarozqueta, Agile and software engineering, an invisible bond, 2020).

The usage of agile methodologies bears relevance to this software industry segment, as well as the deployment of sound software engineering practices as in the local and regional demanding technology markets, as well as customers from the US and Europe that routinely ask provider organizations to present objective proof of their Software Engineering capabilities; in some cases even requiring the adherence to some formal quality model such as ISO9000 (ISO, 2020), SEI-CMMI (Program, 2010) or CoBIT (Elue, 2020) maturity levels, as a condition to compete..

Most scenarios and results captured by the bibliography (Brodman & Jonhson, 1995) reflect the experiences of large-scale organizations formally adopting software engineering practices, leaving smaller ones wondering whether a formal approach is realistic for them, frequently leading to the prior estimation that formal initiatives are simply outside their realm of possibilities. Even though software engineering deployment efforts made at SMEssized companies have already been documented, the focus is often placed on qualitative or methodological factors rather than quantitative ones. It seems that the implicit assumption for software engineering best practices efforts to be a good initiative, unconditionally, regardless of the company's business context is not embraced by many organizations.

At the same time, the deployment of sound software engineering practices is known to be required to stay competitive in terms of productivity, quality, and schedule compliance, especially in the demanding off-shore technology markets.

This notion has been challenged by several authors by whom the actual affordability and suitability of formal software engineering-oriented improvement initiatives for SMEs is questioned from different perspectives.

[1] Work partially funded by grant PID SIUTNCO0004902

Previous work from the authors (Colla, Ruiz de Mendarozqueta, & Bustos, Agile in practice, a systemic approach, 2020) described a comprehensive framework that helps in the understanding of organizations attempting to implement software process improvement (SPI) initiatives and allows to understand the different organizational parameters involved in the business decision, the outcome that might be expected, and the level of risk associated with it.

Until now, research work had to be done using sources from different development markets and results extrapolated to Argentina under the implicit assumption of validity. Very few sources previously addressed a systematic analysis or provide insights into this market.

This paper proposes a contribution by analyzing data collected in Argentina, trying to understand the dynamic behavior of the different variables associated with the usage of agile methods and software engineering practices, to understand the bonds between both approaches in order to evaluate possible strategies to address the likelihood of its results. Finally, threats to the validity of this approach and preliminary conclusions are explored.

2 RESEARCH METHOD

Most empirical studies regarding the adoption of agile methods and software engineering practices rely on very simple statistical and analytical methods such as percentage tables, charts, and related univariate and bivariate statistics. Sample sizes are typically small, with correspondingly few degrees of freedom to support multivariate analysis. A good deal of insight can be gained from relatively simple analytical techniques. Even fewer sources throw light onto the understanding of how organizations address the connection between agile methods and the underlying connection with the software engineering best practices implemented with them.

2.1 Research questions

Our research questions were:

- Are the adoption of Agile methodologies and the embracement of software engineering practices perceived as mutually related by the organizations?
- How the adoption of agile methodologies and deployment of software engineering practices are related to the organizational size and age?
- What is the influence on the adoption of agile methodologies and/or software engineering practices related to the markets the organizations participate in, the deployment of formal quality models evaluation and the operation under incentive programs? In particular how do both correlate to Argentina´s software promotion law (Ley 25922)?

2.2 Data Source

The scope of the collected survey attempts a research activity to include a group representing a variety of software organizations in Argentina. It is composed by few questions related to several organizational characteristics, context factors, and the usage of both agile methods and software engineering practices. A combination of Yes/No, Multichoice, and 5-Likert categorical values are captured through the questions. The survey went public thru different social media, personal network, and other academic channels and enough answers were collected, according to the design of the research experiment, to ensure that the results would be within a pre-defined sample error and confidence interval.

2.3 Analysis Framework

The organization size, measured as the direct software development resources, is mapped as a token of the organization's strength in terms of scale at the moment to decide whether or not to perform investments on improving their performance. The organization age is used as a direct indicator for the room to collect feedback from customers experience, and actual results, into the need to introduce structural compliance with software process methodologies.

Some factors are subject to decisions being made by the management whose relationship to the agile and/or software engineering practices is to be evaluated. Among these factors, the actual core business and the nature of the served markets, might define the need for the organization to raise software development performance level. Other parameters are the management decision to embrace formal quality-related evaluations and the

affiliation to external programs that might be related to the fulfillment or adoption of industry frameworks. Colla et.al (Colla & Montagna, 2008) referred a research made that predicts a significant relationship between the *organization size* (N) and the likelihood of embracing formal quality models. Intuitive as it might seem, this notion had received little attention in published papers in terms of validation.

Finally, the analysis includes the adherence to Argentina's Software Law (agencia.mincyt.) (Ley 25922) as a factor assumed to facilitate the formal adoption of quality systems, and therefore included as a parameter whose relationship with other research factors needs to be explored.

2.4 Design of experiment

Although a full census would be desirable to understand the subjects under research, this is deemed impractical as a preliminary evaluation of the factors addressed by this paper. This might even be impossible, as many organizations would refuse to go public with their internal data in fear of exposing competitive information of internal nature. Because of that, a sample survey has been attempted with a pre-defined level of representation of the target organizations, which derives on a measurable confidence interval on the results. The sample could be considered, in broad terms and not completely void of skew factors, a random one, as the call for answers was made public and no individual answers were solicited. After saying that, the affiliation and personal network of the authors play a role that might, up to some extent, skew the results. However, the resulting dataset collected is deemed acceptable as it reaches the sampling error defined for the experiment.

For the analysis's sake, generalizations would be made with the collected information assuming that a random data sample has been collected, and understanding the threat to validity that this approach might introduce.

According to the data made available by CESSI (OPSSI, 2016), close to 650 organizations are involved in the software development business in Argentina, delivering to different segments and capabilities. This probably would be a very conservative number as many organizations might not be truly devoted to software development but other activities of the value chain of the software industry; however, assuming a larger than needed number, would make the results stronger in terms of the confidence level.

In order to achieve a given significance of the results, it is necessary to identify what would be the minimum sample size, this factor defines both the precision and the confidence interval of the results. It is a judgment call of the authors to balance the precision to be achieved, against the resources realistically available to perform the data collection.

Cochran (Cochran, 1977) recommends a sample size (n_0) for a very large population as:

$$n_0 = \frac{Z^2 \times p \times q}{e^2}$$

Equation 1

Where the normalized random variable (Z), represents the value at the confidence level assuming a normal distribution. Using a value of 1.96 to achieve a 95% confidence level. The assumed proportion of the population with a given attribute (p) and the lack of it (q) is assumed, in the worst case, by assigning the same value to both (0.5). The *minimum sample size* (n_0) can be computed for different *accepted error levels* (e) as seen in Table 1.

			n_0		
		CI	70%	95%	99%
e	ζ	Z	0,7	1,96	2,58
5%	95%		49	384	666
10%	90%		12	96	166
15%	85%		5	43	74
20%	80%		3	24	42
25%	75%		2	15	27
30%	70%		1	11	18
35%	65%		1	8	14
40%	60%		1	6	10
45%	55%		1	5	8
50%	50%		0	4	7

Table 1 Confidence levels for a random variable Z and resulting sample size (Yamane, 1967) (Cochran, 1977)

Based on Table 1, a minimum sample size of 15 answers and a desirable level of 24 answers are adopted as a preliminary design level for the response to the survey that would yield acceptable results for the scope of this effort.

As the total population is finite, a validation is needed to find whether that number is appropriate.

Yamane (Yamane, 1967) provides criteria to define the sample size for small populations, when the sample size might be comparable to the total population or in any case, it cannot be considered as much larger. A simplified formula with maximum variation and 95% confidence interval is given by:

$$n_r = \frac{N_u}{1 + N_u \times e^2}$$

Equation 2

The accepted error level (e) is now defined as to how close the estimations made using the collected data are compared with those of the population it samples. The relation between precision and sample error is given by:

$$Error = 1 - e$$

Equation 3

N	650			
	e	ζ		n_0
	5%	95%		248
	10%	90%		87
	15%	85%		42
	20%	80%		24
	25%	75%		16
	30%	70%		11
	35%	65%		8
	40%	60%		6
	45%	55%		5
	50%	50%		4

Table 2

Applying this criterion, the resulting recommended sample size is given by Table 2where a minimum number of 16 answers and a desired number of 24 answers is obtained, which is deemed as pretty consistent with the previous evaluation. The overall assumed accepted error level might look like a little high, but consideration needs to be given to the fact that this research is aimed to obtain preliminary insights on a previously unexplored subject, and that the authors consider that this sort of precision is a reasonable balance between the available resources and the robustness of the conclusions made possible by them

3 SURVEY DESIGN

Two factors represent the dependent variables under study, the *degree of agile deployment* (AGILE, Y_1) and the *degree of software engineering practices deployment* (SWE,Y_2). Both are captured as categorical variables represented using a 5-Likert scale, where the minimum level is little or no implementation, and the maximum is full adoption, whereas the mid-scale represents the awareness and some fair level of usage. Both scales are designed to represent a similar depth of adoption per level. Organizational characteristics are assigned as independent variables. Organizational size (X_1) and Organizational age (X_2) are both assigned with 5-Likert categorical values. For the size, the CESSI, (OPSSI, 2016), usual categorical scale is used while for the organizational age, an experimental sequence is adopted. The main goal of the organization is based on development type performed. Markets served, quality accreditations achieved, and technology focus, are also captured with multi-choice options, that can be manipulated as different kinds of discrete answers with convenient grouping. The main factors used can be seen at Appendix – Survey Design.

3.1 Survey Design and distribution

As design criteria, the total survey has created as a "*one-pager*" to increase the likelihood of being answered (Mardsen & Wright, 2010). A small operating definition is attached to each question and general instructions for fulfilling and returning are provided as well. A confidential statement ensures the participant that no individual answer will be used or published, all the results would be statistical aggregates characterizing the sample to understand the whole population. Fulfillment help is provided in terms of drop lists and checkboxes to uniform the answers provided within the defined categories. Google Forms, (Ruiz de Mendarozqueta, Goggle Forms), has been used to implement the survey form and several validations and verification tests were performed by the authors to ensure the functionality of different options.

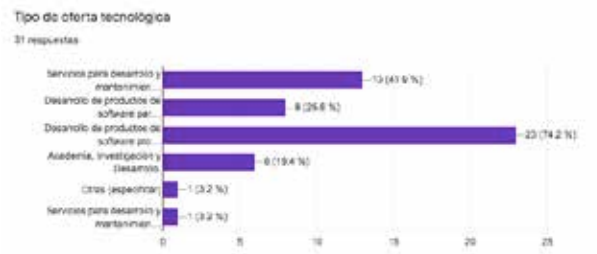
The survey was published on the *LinkedIn* account (Ruiz de Mendarozqueta, Linkedin) and other social media platforms for all the authors. A fair amount of bouncing from direct network professionals was observed allowing the survey to reach a larger audience resulting in the request to reach several hundred individual practitioners at the end of the diffusion process.

4 SURVEY ANALYSIS

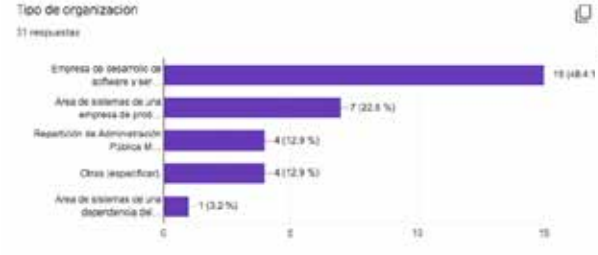
4.1 Survey demographics

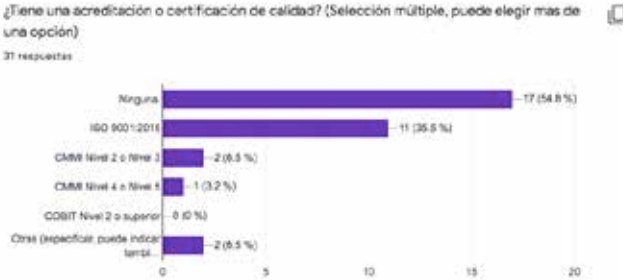
A total of 30 valid and unique responses were provided as collected by the Google Forms tool. The distribution of organizational size and by organizational age is given by the following figures.

The technology area where the organizations perform and the markets they serve are shown in the following figures:



The organization type characterized by type and the kind of quality accreditation they choose are shown in the following figures:





As per the subject of interest for the survey, the agile methodologies and the perception of best practices adoptions is shown in the following figures.

As a validation of the distribution of organizational size, the proportions devoted to different markets the mainline of activity have been compared with the 2018 CESSI sponsored survey [(CESSI, 2018)] , the main proportions are found within reasonable approximation to the survey results as a sample of this population.

4.2 Evaluation of dependent variables

The main tools for statistically analyzing a dataset, differ depending on whether the distribution of the data follows a normal distribution or not. For non-normal distributions, so-called "non-parametric" tools are used, which, in general, are less powerful and versatile. It is therefore an accepted practice to use tools aimed at normal distributions, even in cases where the distribution differs from it to a lesser extent. The organization size is found not to follow a normal distribution since the Anderson-Darling normality test has a p-value=0.005. The organization age (AGE) normality test has also a p-value=0.005 and doesn't follow a normal distribution either.

Assumed both dependent variables represent equivalent levels of implementation for both agile practices and software engineering practices, the Mann-Whitney test compares the sample medians to be equal vs. not equal, resulting in a p=0,7958 therefore, the null hypothesis cannot be rejected and both populations can be considered as having the same median value. The paired t-test needs to be used with caution because of the lack of normal distribution on both variables but it yield a T-Test of mean difference = 0 (vs ≠ 0): P-Value = 0,889 therefore the null hypothesis of no mean difference cannot be rejected. Using Ordinal Logistic Regression, (Kruskal, 1954), an evaluation on the relationship between the dependent variables with both organizational size (N) and organizational age (AGE) is made, a result of p>0.05 means there is insufficient evidence to claim the model does not fit the data adequately, and therefore, the variables are related as seen in the following tables.

	Y(AGILE)	Y(SWE)
N	**0.435**	0.183
AGE	0.12	**0.948**

	AGILE	SWE
GLOBAL	**0.062 (yes)**	0.244 (not)
SPI	0.604 (not)	**0.007 (yes)**
EXT	0.104 (yes)	0.322 (not)

The impact of parameters such as the market being served (GLOBAL), the adoption of quality systems certification/assessments (SPI), and the operation under external program (EXT), is evaluated in terms of the dependency of the agile or software engineering practices adoption with them using a Chi-Square method (Table 3).

Source	p-value
N	0.667
AGE	**0.032**
GLOBAL	0.473
SPI	0.199
EXT	0.270

Table 3 Generalized Linear Model between independent variables to the adoption of agile methodologies

Using a Generalized Linear Model (GLM) regression between the independent variables and parameters, and the adoption of agile methodologies, can be also seen in Table 4, where a p-value of less than 0.1 means a dependency was found, whilst a larger p-value indicates the independence (null hypothesis), cannot be rejected. Repeating the analysis, but now with the implementation of software engineering practices, can be seen at

Source	p-value
N	**0.060**
AGE	0.487
GLOBAL	0.877

SPI	**0.079**
EXT	0.474

Table 4 GLM between independent variables to the adoption of software engineering practices

Finally, the discretized responses representing agile methodologies and software engineering practices adoptions, are related to the adoption of Argentina´s software promotion law as a discrete (binary) variable. The Chi-Square analysis between SWE and LEY yield p=0.033, so a dependency has been found while the relationship between AGILE and LEY yield p=0.783 and a dependency has not been found.

5 DISCUSSION

The adoption of agile and software engineering are similar in organizations, the higher the one the other correlates as higher too. This is a hint regarding that organizations apply stricter agile methods as they are aware of the need to deploy software engineering practices as well. Organization size dominates the adoption of agile methodologies, in a stronger way than the adoption of software engineering practices, whilst the opposite is suggested for software engineering practices.

The operation servicing global markets is related to the adoption of agile methodologies, whilst the adoption of strict, committed or certified, quality frameworks is related to the adoption of software engineering practices, surprisingly, the usage of external incentive programs seems to relate stronger with agile than the adoption of software engineering practices. The operation under the benefits of Argentina's Software Law drives the adoption of software engineering practices but it's not related to the usage of agile methodologies within the statistical margin assumed.

Threats to validity

The data for this study were collected using a survey instrument designed to support this particular research study, a realistic assessment of the number of feasible answers to be collected, results on the authors' decision to accept a relatively high sample error. This has resulted in more threats to internal validity than would be desirable. We have persevered because this is very scarce data that sheds light on important, but previously unaddressed questions, especially for the Argentina environment.Organizations might be tempted to give inaccurate answers that reflected better on them than their real reasons, especially when talking about factors related to their current maturity of adoption of agile and software engineering practices. This threat is partly controlled by a careful survey design, avoiding negative implications of the different levels used by the answers.

In an idealized setting, organizational decisions are cost vs. benefit judgments; however, in practical terms, this is a simplistic theoretical perspective that might be invalid for any individual organization to decide on adopting a given methodology. This is especially true for the resources-constrained small organizations, which were a large part of the surveyed population. A further complication is that the independent variables were assessed using retrospective recall. This involves the risk of the introduction of retrospective bias. The method used to distribute and gather data, might introduce some skewness in the conclusions because of being conducted thru the professional network of the authors; however, the level of bouncing beyond that network, the random nature of the respondents that choose to answer, and additional validations performed with the data, give the authors some confidence that this factor is probably within the overall error level accepted for the conclusions. In this regard, some results were accepted with a p-value of 0.05 or below whilst for other a value of 0.1 was found acceptable; in general, for non-parametric methods, the latter was found acceptable given the overall lack of power of the test itself.

Regarding external validity, it is possible that our findings do not generalize to other contexts. Although the sample is large enough, there may be hidden systematic bias affecting the valid generalization of results. Organizations in the study are all Argentineans, which was one of the focus of the work, given the lack of previous analysis in the same direction, but this makes it worth note that our findings might not be generalized to other environments. However, it is relevant to highlight that the overall results agree with a large body of research and industry references, and our previously published modeling based on simulation techniques.

6 Conclusion

The survey analysis provides some findings of the adoption of agile methodologies together with software engineering practices at the Argentina software organizations, and extends previous research from several perspectives. Of course, all conclusions from a single study are tentative at best, and require confirmation through further research, since additional and new questions are identified. In the following points, we summarize the main conclusions related to the original research questions addressed:

- Are the adoption of Agile methodologies and the embracement of software engineering practices perceived as related by the organizations?

  Yes, they are related, a statistically significant correlation has been found within the accepted error margins.
- ¿How the adoption of agile methodologies and deployment of software engineering practices are related to the organizational size and age?

  Both factors are related to organizational size and age. Agile is related to organization age whilst software engineering practices are related to size.
- What is the influence on the adoption of agile methodologies and/or software engineering practices related to the markets the organizations serve, the deployment of formal quality models evaluation, and the operation under incentive programs? In particular, how both correlate to each identified incentive program?

  Agile is related to the serving of global markets, while software engineering is related to the adoption of formal software models.

7 Future work

Methodological implementations and the embracement of software engineering practices are controversial issues and more research is needed to study their relationship. Several levels of analysis are possible (i.e. individual, group, process, and organization, each one with complex interactions with the others). Further research should be related to the study of the improved analysis of efforts being conducted in the same environment, comparison of key investment characteristics, and validation of the results. At the same time, context variables, such as organizational culture, business environment and other factors key to the survival of SMEs did not play an important role in understanding the relationship with outcomes available in the bibliography. It is needed then to investigate further the importance of such variables to several types of the usage of common agile methodologies within the framework of accepted software engineering practices and to validate the approaches proposed for solving them. We would like to express our thanks to the Universidad Tecnológica Nacional (Regional Córdoba) for partially fund the publication of this paper as part of the research program up to this point.

8 Bibliography

agencia.mincyt. (n.d.). *agencia.mincyt.* Retrieved 10 17, 2020, from Ley de Software: http://www.agencia.mincyt.gob.ar/upload/ley_25922-2.pdf

Agile Alliance. (2020, 08 12). *Agile Practices Timeline*. Retrieved from Agile Practices: https://www.agilealliance.org/agile101/practices-timeline/

Akbar, R. (2019). Tailoring Agile-Based Software Development Processes. *IEEE Access, 2019*.

Alegrìa, J. H., & Bastarrica, M. (2007). *Implementing CMMI using combination of Agile methods.* V9(N1).

Appleton, B., Berczuk, S., & Cowham, R. (2005). *The Agile Difference for SCM*. Retrieved from CrsossRoads: https://www.cmcrossroads.com/article/agile-difference-scm

Banerjee, A., Narasimhan, B., & Kanakalata, C. (2011). *Experience of Executing Fixed Price Off-shored Agile Projects.* Proceedings of the 4th India Software Engineering Conference. ACM.

Beck, K., & Boehm, B. (2003). Agility through Discipline: a debate. *June 2003*.

Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Principles behind the Agile Manifesto*. Retrieved from http://agilemanifesto.org/principles.html

Bhasin, S. (2012). Quality Assurance in Agile –A study towards achieving excellence. pp. pp 64-67.

Bick, S., Spohrer, K., Hoda, R., Scheerer, A., & Heinzl, A. (2018). Coordination Challenges in Large-Scale Software Development: A Case Study of Planning Misalignment in Hybrid Settings. *IEEE Transactions on Software Engineering, Year: 2018, Volume: 44, Issue: 10*.

Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *The Journal of Political Economy, V81*(N3), pp. pp 637-654.

Boehm, B. (1987). *Improving Software Productivity.* IEEE Software.

Boehm, B., & Basili, V. R. (2001). Software Defect Reduction Top 10 List. *IEEE Computer, January 2001*.

Boehm, B., & Turner, R. (2005). Management Challenges to Implementing Agile Processes in Traditional Development Organizations.

Brealey, R., & Myers, S. (2016). *Principles of Corporate Finance 12th Edition.* McGraw-Hill, 6th Edition.

Brodman, J., & Jonhson, D. (1995). Return on investment (ROI) from software process improvement as measured by US industry. *Software Process: Improvement and Practice,*.

Bustard, D., Wilikie, G., & Greer, D. (2013). The Maturation of Agile Software Development Principles and Practice: Observations on Successive Industrial Studies in 2010 and 2012. *20th Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems (ECBS)*.

Caballero, E., Calvo-Manzano, J., & Feliu, T. S. (2011). Introducing Scrum in a Very Small Enterprise: A Productivity and Quality Analysis. pp. pp. 215-224.

CESSI. (2018). *Reporte anual del sector de software y servicios informáticos de la República Argentina.* Retrieved from http://cessi.org.ar: https://www.cessi.org.ar/descarga-institucionales-2330/documento2-290a5d80ccf2da8c5d3935209aa10568

Chora, M., Springer, T., Kozik, R., López, L., Martínez-Fernandez, S., Ram, P., . . . Franch, X. (2020). Measuring and Improving Agile Processes in a Small-size SoftwareDevelopment Company. *IEEE Access (Volume: 8), 2020*.

Clark, B. (2000). Quantifying the effects of Process Improvement on Effort. *IEEE Software. Nov 2000*.

Cochran, W. G. (1977). *Sampling Techniques, 3rd Edition.* Wiley.

Cockburn, A. (2007). Agile Software Development. , Addison-Wesley.

Cockburn, A., & Highsmith, J. (2001). Agile Software Development: The People Factor. *IEEE Computer Year: 2001, Volume: 34*.

Cohen, D., Lindvall, M., & Costa, P. (2004). An Introduction to Agile Methods. *ADVANCES IN COMPUTERS, VOL. 62*.

Cohn, M. (2010). *Succeding with Agile.* Addison Wesley.

Cohn, M. (2012). *Essential Scrum.* Adisson Wesley.

Coleman, D. (2015). *El cerebro y la inteligencia emocional: Nuevos descubrimientos.* Penguin Random House Grupo Editorial España.

Colla, P. (2012). *Marco para evaluar el valor en metodología SCRUM.* La Plata-Argentina.: 13th Argentine Symposium on Software Engineering.

Colla, P. (2016). *Uso de opciones reales para evaluar la contribución de metodologías KANBAN en desarrollo de software.* Tres de febrero: SADIO ISSN: 2451-7593.

Colla, P. E., Bustos, F. O., & Ruiz de Mendarozqueta, A. (2020). Agile and software engineering, an invisible bond. *ASSE 2020.* Buenos Aires: Sadio-JAIIO.

Colla, P. E., Ruiz de Mendarozqueta, A., & Bustos, F. O. (2020). Agile in practice, a systemic approach. *ASSE 2020.* Buenos Aires: Sadio - JAIIO.

Colla, P., & Montagna, M. (2008). Software Process Improvement Behavior of Small and Medium Organizations in Argentina. *CACIC'08* . Chilecito: CACIC'08 .

Davis, A. (1994). *FIFTEEN PRINCIPLES OF SOFTWARE ENGINEERING.* IEEE.

Davis, A., Bersoff, E., & Comer, E. (1988). *A Strategy for Comparing Alternative Software Development Life Cycle Models.* IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 14, NO. IO,.

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). *Scrum Primer.* Retrieved 05 31, 2020, from https://scrumprimer.org/

DeMarco, T., & Lister, T. (1987). *Peopleware.* Dorset House.

Dingsøyr, T., Fægri, T. E., Dybå, T., Haugset, B., & Lindsjørn, Y. (2016). Team Performance in Software Development Research Results versus Agile Principles. *IEEE Software (Volume: 33 , Issue: 4 , July-Aug. 2016)*.

Duncan, S. (2019). *Understanding Agile Values & Principles.* C4Media, InfoQ.com.

Ebert, C., & Paasivaara, M. (2017). Scaling Agile. *IEEE Software (Volume: 34 , Issue: 6 , November/December 2017)*.

Elue, E. (2020, 10 17). *ISACA.ORG*. Retrieved 10 17, 2020, from Effective Capability and Maturity Assessment Using COBIT 2019: https://www.isaca.org/resources/news-and-trends/industry-news/2020/effective-capability-and-maturity-assessment-using-cobit-2019

Endres, A., & Rombach, D. (2003). *A Handbook of Software and Systems Engineering.* Pearson Addison Wesley.

Fairley, R., & Bourque, P. (2014). *SWEBOK v 3.0 Guide to the Software Engineering Body of Knowledge.* IEEE Computer Society.

Falessi, D., Cantone, G., Sarcia, S. A., Calavaro, G., Subiaco, P., & D'Amore, C. (2010). Peaceful Coexistence: Agile Developer Perspectives on Soft-ware Architecture. *IEEE Software Year: 2010, Volume: 27, Issue: 2*.

Fritzche, M., & P.Keil. (2007). Agile Methods and CMMI: Compatibility or Conflict ?

Gilb, T. (1988). *Principles of Software Engineering Management.* Addison-Wesley.

Glass, R. (2002). *Facts and Fallacies of Software Engineering.* Addison Wesley.

Glazer, H., Dalton, J., Anderson, D., Konrad, M., & Shrum, S. (2008). *CMMI or agile: why not embrace both!* SEI TECHNICAL NOTE.

Goldenson, D., A.Liu, & Jianping, Q. (2006). *CMMI-Based Process Improvement: How and When Does Success Happen?* CMMI Technology Conference: Software Engineering Institute.

Good, J. M. (2003). *A Pragmatic Approach to the Implementation of Agile Software Development Methodologies in Plan-Driven Organisations (MSc Thesis).* Lincoln University.

Hallowell, D.L. (2003). *Six Sigma Software Metrics Maturity*. Retrieved 2019, from iSixSigma: https://www.isixsigma.com/industries/software-it/exploring-defect-containment-metrics-agile/

Harvie, D., & Agah, A. (2016). Targeted Scrum: Applying Mission Command to Agile Software Development. *IEEE Transactions on Software Engineering (Volume: 42 , Issue: 5 , May 1 2016)*.

Hoda, R., Salleh, N., & Grundy, J. (2018). THE RISE AND EVOLUTION OF AGILE SOFTWARE DEVELOPMENT. *IEEE Software (Volume: 35 , Issue: 5 , September/October 2018)*.

Hohmann, L. (1997). *Journey of the Software Professional* . Prentice Hall.

Hummel, O., & Burger, S. (2013). A pragmatic means of measuring the complexity of source code ensembles.

Humphrey, W. S. (1989). *Managing the Software Process.* Addison-Wesley.

Hung, M., & So, L. (2010). The Role of Uncertainty in Real Option Analysis.

Institute, S. (n.d.). *Scrum Institute*. Retrieved 06 08, 2020, from https://www.scrum-institute.org/inspect-and-adapt-scrum-framework.php

ISACA. (2018). COBIT 5 Framework. In ISACA.

Ismail, N. (2016). *UK wasting 37 billion a year on failed agile IT projects*. Retrieved from https://www.information-age.com/uk-wasting-37-billion-year-failed-agile-it-projects-123466089/

ISO. (2020, 06 08). *ISO 9000:2015*. Retrieved from https://www.iso.org/obp/ui/es/#iso:std:iso:9000:ed-4:v1:es

Johnson, H., & Sims, C. (2012). *Scrum: a Breathtakingly Brief and Agile Introduction.* Dymaxicon.

Jorgensen, K. M. (2003). A review of software surveys on software effort estimation. *Proceedings ISESE 2003.* , (pp. pp-223-230). Rome, Italy.

Jorgensen, M. (2019). Relationships Between Project Size, Agile Practices, and Successful Software Development Results and Analysis. *IEEE Software Year: 2019 Volume: 36, Issue: 2*.

Karlstrom, R. (2005). Combining agile methods with stage-gate project management. *IEEE Software, Year: 2005, Volume: 22, Issue: 3*.

Kersten, M. (2018). What Flows through a Software Value Stream? *IEEE Software Year: 2018, Volume: 35, Issue: 4*.

Knox, S. (1993). Modeling the Cost of Software Quality. pp. pp 9-16.

Kruskal, L. A. (1954). Measures of Association for Cross Classifications. *Journal of the American Statistical Association*, 732-764.

Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Trektere, K., McCaffery, F., . . . Prause, C. R. (2019). Hybrid Software Development Approaches in Practice: A European Perspective. *IEEE Software (Volume: 36 , Issue: 4 , July-Aug. 2019)*.

Kunz, M., Dumke, R. R., & Zenker, N. (2008). Software Metrics for Agile Software Development. pp. pp. 673-678 .

Lan, C., & Balasubramaniam, R. (2007). Agile Software Development: Ad Hoc Practices or Sound Principles. *April 2007*.

Lawlis, P. K., M., F. R., & B., T. J. (1995). A Correlational Study of the CMM and Software Development Performance. pp. pp. 21-25.

Lee, G., & Xia, W. (2010). TOWARD AGILE: AN INTEGRATED ANALYSIS OF QUANTITATIVE AND QUALITATIVE FIELD DATA ON SOFTWARE DEVELOPMENT AGILITY. pp. pp 87-114.

Liebert, F. (2019). *BARRIERS TO SUCCESSFUL REALIZATION OF NEW PRODUCT DEVELOPMENT PROJECTS IN THE IT INDUSTRY.* Silesian University of Technology, Faculty of Organization and Management.

M. Staples, M. R. (2007). An exploratory study of why organizations do not adopt CMMI. *The Journal of Systems and Software 80* , p.p. 883–895.

Mahnic, V. (2012). A Capstone Course on Agile Software Development using SCRUM. *IEEE TRANSACTIONS ON EDUCATION, VOL. 55, NO. 1, FEBRUARY 2012*.

Maller, P., C.Ochoa, & Silva, J. (2004). Lightening the software production process in a CMM level 5 framework. *IEEE Latin American Transactions, V3(N1)*(pp 15-22).

Mantovani Fontana, R., Reinehr, S., & Malucelli, A. (2015). *Agile Compass: A Tool for Identifying Maturity in Agile Software Development Teams.* IEEE Software (Volume: 32 , Issue: 6 , Nov.-Dec. 2015).

Marcal, A., DeFreitas, B., Furtado, F., & Belchior, A. (2008). Blending SCRUM practices and CMMI Project Management Process Areas. *Innovation System Software*(pp 18-29).

Mardsen, P., & Wright, J. (2010). *Handbook of Survey Research.* Emerald.

Martin, R. (2012). *Código limpio: Manual de estilo para desarrollo ágil de software.* Anaya.

Martin, R. (2019). *Clean Agile: Back to Basics.* Prentice Hall.

Matson, J., Barrett, B., & Mellichamp, J. (1994). Software development cost estimation using function points. *20.4, pp. 275-287.*

McConnell, S. (1993). *Code Complete.* Microsoft Press.

McConnell, S. (1996). *Rapid Development.* Microsoft Press.

McConnell, S. (2019). *More Effective Agile: A Roadmap for Software Leaders.* Construx Press.

McMahon, P. (2010). *Integrating CMMI and Agile Development.* Addison-Wesley Professional;.

Meadows, D. (2008). *Thining in Systems: a primer.* Chelsea Green.

Mendarozqueta, A. R., & Andriano, N. (2014). Un enfoque para la mejora continua basado en los principios ágiles.

Miller, G. (2013). Agile problems, challenges, & failures. *PMI® Global Congress 2013*, pp. pp.1-8.

Mohan, K., Ramesh, B., & Sugumaran, V. (2010). Integrating Software Product Line Engineering and Agile Development. *IEEE Software (Volume: 27 , Issue: 3 , May-June 2010)*.

Morse, L. (2012). *3 Paradigm Shifts of Agile*. Retrieved 05 04, 2019, from Solutions IQ: https://www.solutionsiq.com/resource/blog-post/3-paradigm-shifts-of-agile/

Mukker, A., Mishra, A. K., & Singh, L. (2014). Enhancing Quality in Scrum Software Projects. pp. pp 682-688.

Mun, J. (2002). *Real Options Analysis, Tools and Techniques for Valuing Strategic Investment and Decisions.* Hoboken, New Jersey: John Wiley & Sons.

O'Regan, G. (2017). *Concise Guide to Software Engineering.* Springer.

OPSSI. (2016). *Reporte anual sobre el Sector de Software y Servicios Informáticos de la República Argentina.* Buenos Aires: CESSI.

Paulk, M. C. (2002). *Agile Methodologies and Process Discipline.* Institute for Software Research. Paper 3.

Poppendieck, M., & Poppendieck, T. (2003). *Lean Software Development: An Agile Toolkit.* Addison Wesley.

Program, S. E. (2010). *CMMI® for Development, Version 1.3.* SEI.

Rafaela Mantovani Fontana, S. R. (n.d.).

Rico, D. F. (2008). What is the ROI of Agile vs. Traditional Methods? pp. pp. 9–18.

Ruiz de Mendarozqueta, A. (n.d.). *Goggle Forms*. Retrieved 17 10, 2020, from https://docs.google.com/forms/d/1c0nhdf7hENrL9_r7mD0mUPQfOiIjOvD5FdJ2mcU9WOM/edit

Ruiz de Mendarozqueta, A. (n.d.). *Linkedin*. Retrieved 10 17, 2020, from Linkedin: https://www.linkedin.com/posts/alvarordm_encuesta-sobre-agilidad-y-pr%C3%A1cticas-de-ingenier%C3%ADa-activity-6701246489215168512-0BgA/

Ruiz de Mendarozqueta, A., Bustos, F., & Colla, P. (2019). *Agile in practice, a systemic approach.* Paper accepted for 48 JAIIO-ASSE 2019, to be published in 49 JAIIO-ASSE 2020.

SAFe. (n.d.). Retrieved 06 08, 2020, from https://www.scaledagileframework.com/agile-architecture/

Sandu, I., & Salceanu, A. (2018). New approach to agile cycles containment effectiveness metrics in automotive software development. pp. pp. 3-8.

Sargent, R. (2009). Verification and validation of simulation models. *Proceedings of the 2009 Winter Simulation Conference, ed. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls,*.

Sauer, J. (2005). *Agile Practices in Offshore outsourcing- An analysis of published experiences.* ECSCW 2005.

Schwaber , K., & Sutherland, J. (2017). *Scrum.org.* Retrieved 06 31, 2020, from The home of Scrum: https://www.scrum.org/resources/scrum-guide

Schwaber, K. (2005). *A CIO's Playbook for Adopting the Scrum Method of Achieving Software Agility.* Srum Alliance.

Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide*. Retrieved from Scrum.org

SCRUMstudy. (2013). *A Guide to the SCRUM BODY OF KNOWLEDGE.* Retrieved 05 31, 2020, from https://www.scrumstudy.com/: https://www.scrumstudy.com/

Shore, J., & Warden, S. (2008). *The Art of Agile Development.* O'Reilly.

Shuterland, J., Jakobsen, C., & K.Johnson. (2008). Scrum and CMMI L5 The magic potion for the code warriors. *V*(N).

Sommerville, I. (2015). *SOFTWARE ENGINEERING 10th Edition.* Pearson.

Stellman, A. (2014). *Learning Agile: Understanding Scrum, XP, Lean, and Kanban.* O'Reilly.

Team, C. P. (2010). *CMMI for Development, version 1.3*. Pittsburgh, Pennsylvania, USA: Software Engineering Institute (SEI), November 2010.CMU/SEI-2010-TR-033.

Telemaco, U., Oliveira, T., Alencar, P., & Cowan, D. (2020). A Catalogue of Agile Smells for Agility Assessment. *IEEE Access, Year: 2020, Volume: 8*.

Turner, R., & Jain, A. (2002). Agile meets CMMI: Culture clash or common cause. *XP/Agile Universe LNCS 2418*.

Vallon, R., Strobl, S., Bernhart, M., Prikladnicki, R., & Grechenig, T. (2016). ADAPT A Framework for Agile Distributed Software Development. *IEEE Software (Volume: 33 , Issue: 6 , Nov.Dec. 2016)*.

Vijay, D., & Ganapathy, G. (2014). Guidelines to minimize the cost of software quality in agile SCRUM process. *Vol.5, No.3*, pp. pp 61-69.

Vijayasarathy, L. R., & Butler, C. W. (2016). Choice of Software Development Methodologies – Do Project, Team and Organizational Characteristics Matter? *IEEE Software (Volume: 33 , Issue: 5 , Sept.-Oct. 2016)*.

Vishal, S., & Kishen, I. (2007). *Will Agile Methodologies work in offshore outsourcing?* San Diego, USA: SWDSI07.

Weinberg, G. (1992). *Quality Software Management (Vol 1 Systems Thinking).* Dorset House.

Yamane, T. (1967). *Statistics, An Introductory.* NY: Harper and Row.